Evaluation of an ensemble based 4D Var assimilation

Cordelia Robinson, Yin Yang, Dominique Heitz, Etienne Mémin

Evaluation of an ensemble based 4D Var assimilation

C. Robinson, Y. Yang, D. Heitz, E. Mémin

INRIA Rennes – Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes, France

Irstea, UR TERE, Rennes, France

Aims and model

- Objective :
  - Compare En4DVar with a classic 4DVar method
- 2D shallow water model

  $\partial_t h + \nabla \cdot \mathbf{q} = 0$

  $\partial_t \mathbf{q} + \nabla \cdot (\frac{1}{h}\mathbf{q} \otimes \mathbf{q} + \frac{1}{2} g h^2 Id) = 0$

Incremental 4DVar vs En4DVar assimilation techniques

Incremental 4DVar assimilation

- Cost function using static covariance matrices $\mathbb{B}$ and $\mathbb{R}$

$$J(\delta X_0) = \frac{1}{2}\sum_{\Omega} ||\delta X_0||_{\mathbb{B}^{-1}} + \frac{1}{2}\sum_{0}^{T}\sum_{\Omega} ||\mathbb{H}(X^b) + \partial_X \mathbb{H}(\partial_X \mathbb{M}(\delta X_0)) - Y||_{\mathbb{R}^{-1}}$$

- Adjoint equation determined by TAPENADE

$$-\partial_t \lambda + (\partial_X \mathbb{M}^{\#})\lambda = \sum_{i=1}^{N} (\partial_X \mathbb{H})^{\#} \mathbb{R}^{-1} (Y - \mathbb{H}(X^b))$$

$$\lambda(t_f) = 0$$

- We deduce the gradient

$$\nabla J(\delta X_0) = \lambda(t_0)$$

En4DVar assimilation

- Cost function using flow dependent background error covariance matrix within the context of the preconditioning techniques $\delta X_0 = \mathbb{B}^{1/2}\delta Z_0$,

$$J(\delta Z_0) = \frac{1}{2}\sum_{\Omega} ||\delta Z_0|| + \frac{1}{2}\sum_{0}^{T}\sum_{\Omega} ||\mathbb{H}(X^b) + \partial_X \mathbb{H} \partial_X \mathbb{M} \mathbb{B}^{1/2} \delta Z_0 - Y||_{\mathbb{R}^{-1}}$$

  where the $\mathbb{B}^{1/2}$ matrix estimated from the difference between each ensemble member and ensemble mean

$$\mathbb{B}^{1/2} \approx \frac{1}{\sqrt{N-1}}(X_0^{b,1} - \bar{X}_0^b, ..., X_0^{b,N} - \bar{X}_0^b)$$

- We estimate the evolution of the $\mathbb{B}^{1/2}$ matrix from the evolution of the ensemble fields in observation space

$$\partial_X \mathbb{H} \partial_X \mathbb{M} \mathbb{B}^{1/2} \approx \frac{1}{\sqrt{N-1}}(\mathbb{H}\mathbb{M}X_0^{b,1} - \mathbb{H}\mathbb{M}\bar{X}_0^b, ..., \mathbb{H}\mathbb{M}X_0^{b,N} - \mathbb{H}\mathbb{M}\bar{X}_0^b)$$

- Localization technique used to eliminate sampling error in state space

$$P_b^{1/2} = (\mathbb{C}^{1/2} \cdot \mathbb{B}_1^{1/2}, ..., \mathbb{C}^{1/2} \cdot \mathbb{B}_N^{1/2})$$

- Minimization performed with **LBFGS algorithm**: limited memory quasi Newton method

Results

Synthetic data

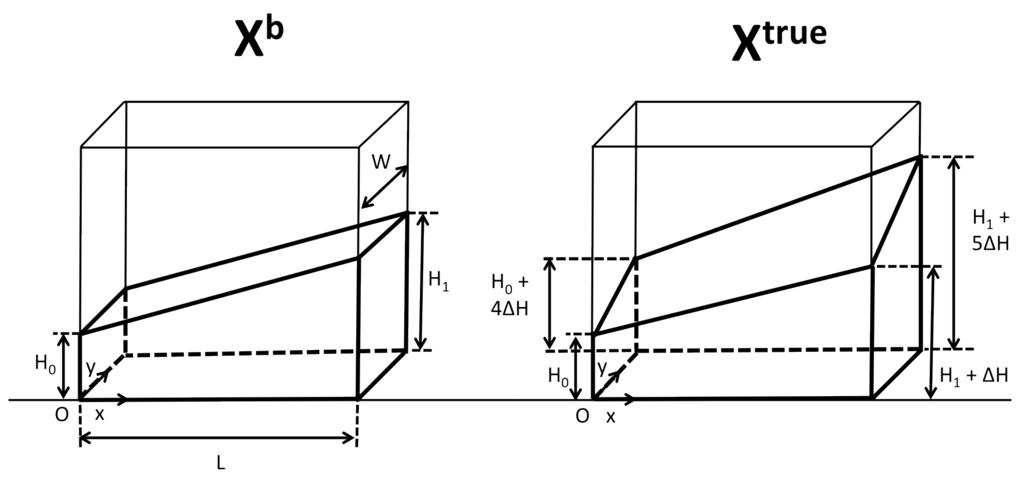

Figure: Background initial state (left) and True initial state (right)

- Background initial condition
  - $L = 25cm$
  - $W = 10cm$
  - $H_0 = 2cm$
  - $H_1 = 7cm$
  - $U(x, y, t_0) = 0$
  - $V(x, y, t_0) = 0$
- Exact initial condition
  - $L = 25cm$
  - $W = 10cm$
  - $H_0 = 2cm$
  - $\Delta H = 0.25cm$
  - $U(x, y, t_0) = \epsilon_0^u(x, y)$
  - $U(x, L, t_0) = \epsilon_0^u(x, L) + 1cm.s^{-1}$
  - $V(x, y, t_0) = \epsilon_0^v(x, y)$
  - $V(W, y, t_0) = \epsilon_0^v(W, y) + 1cm.s^{-1}$

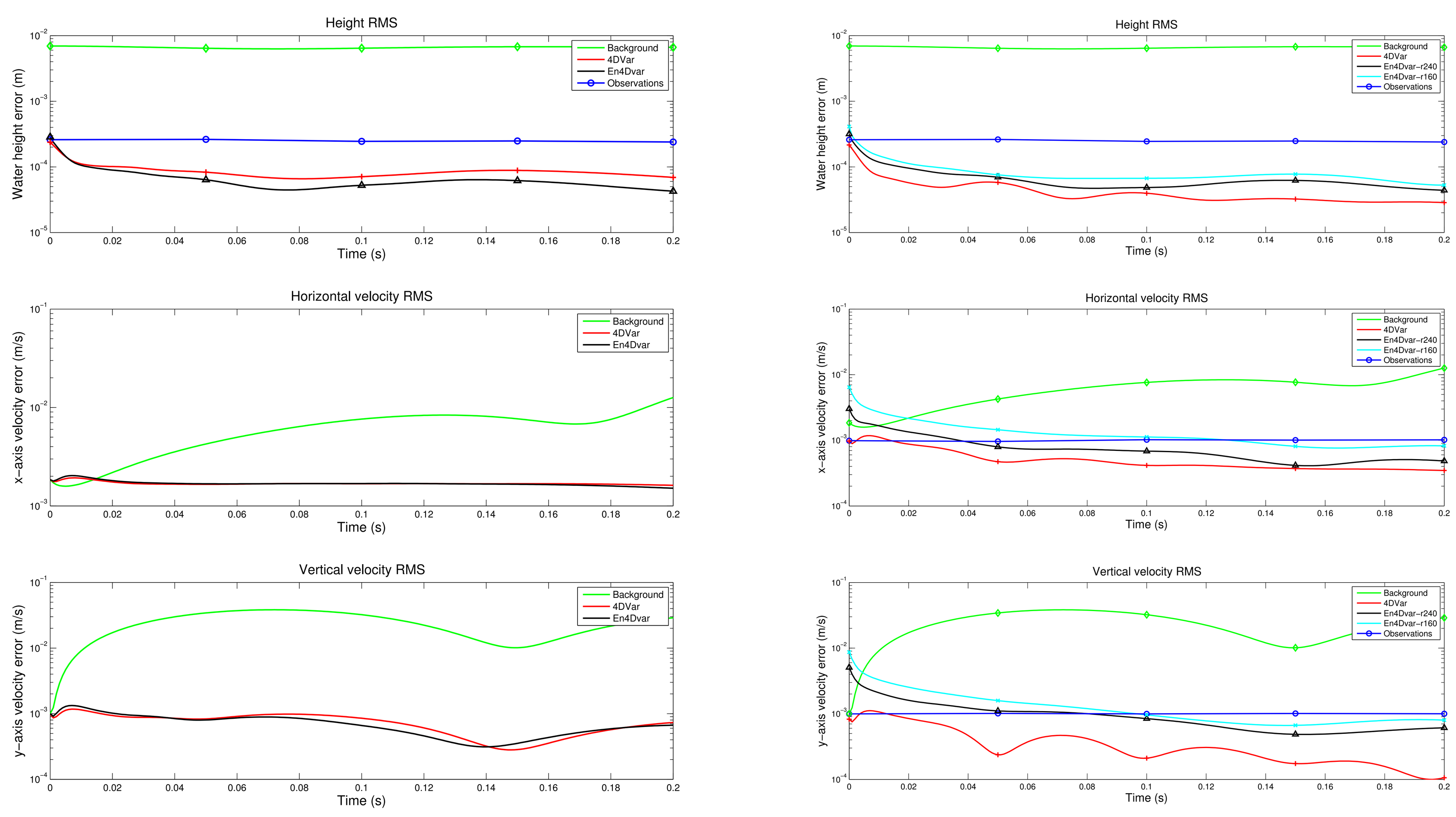

Figure: RMS semilog of the assimilation with the height observations only (left) and both height and velocity observations (right)

- Assimilation with height observations only
  - En4DVar is slightly better than 4DVar
  - Same computation time
  - En4DVar requires more memory space
- Assimilation with height and velocity observations
  - En4DVar requires much more computation time
  - En4DVar leads to better results with a higher truncation mode
  - Higher truncation mode demands higher computation time and memory needs

Experimental data

We only possess the height observations given by the depth sensor (Kinect sensor)

Figure: Experiment configuration with the depth sensor (Kinect sensor)

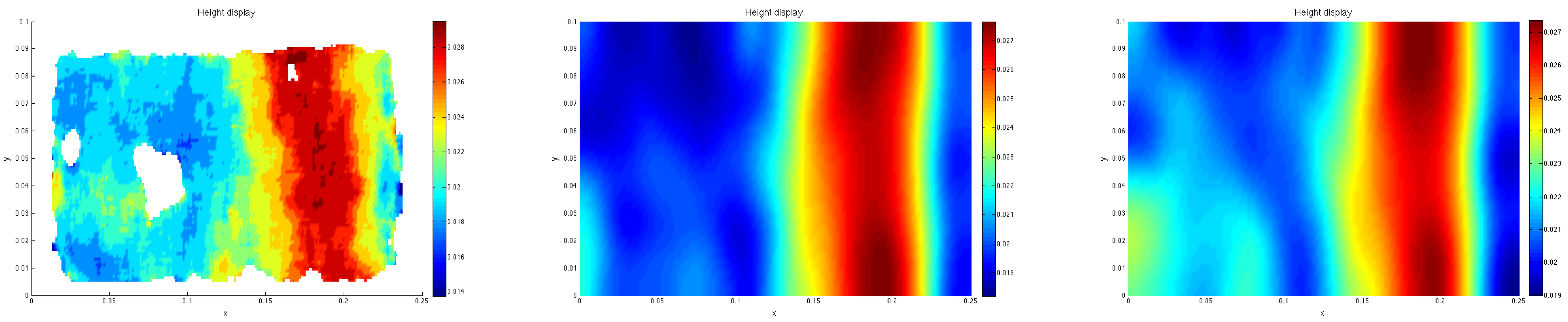

Figure: From left to right: free surface height obtained by the kinect, 4DVar and En4DVar at t=0.06s

Conclusions

- Sensibly the same computational time cost, En4DVar yields better results than the classic 4DVar assimilation when we have only height observations
- En4DVar is easy to implement for any given model. We gain a lot of time with the parallelization computing technique. En4DVar implemented with only one outer loop iteration and needs about 100 iterations for the optimization. Requires a lot of memory.
- 4DVar requires the tangent and adjoint operators. The assimilation converges with 3 outer loop iterations and requires less inner loop iterations for the optimization.

http://irisa.fr/fluminance – {cordelia.robinson, yin.yang, dominique.heitz, etienne.memin}@inria.fr